Remarks

This amendment is in response to the Office Action mailed April 5, 2007. The many helpful suggestions in the Office Action are highly appreciated and have thereafter been incorporated in the amended claims.

Claim Rejections - 35 USC 112

Claims 1-15 are rejected under 35 U.S.C 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

Claim 1 recites "without loading and rebooting" which contradict applicant's specification; and the language "system RAM" is unclear.

Claims 1-19 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 1 recited a method of "allowing", conflict with the remainder of claim which recites specific steps for "causing" a mobile device to execute a second operating system.

Claim 2 applicant's parenthetical insertion of "data and code" renders the limitation unclear. It is unclear how the MMU could manipulate memory protection attributes for the data and code stored in the memory.

Claim 3 is unclear what constitutes the "top" of physical memory.

Claim 4 contains typographical errors - "further comp", "of more".

Claim 6 is unclear what makes a register "special".

In Claim 16 is unclear what is being claimed, unclear whether an "inter code-segment invocation" is an action being performed (since the aforementioned appending steps "allows" the invocation) or a code/instruction in the code segment (since it is pre-linked to a jump instruction).

After reviewing the listed items by very detailed work of the examiner, and also with a telephone interview with the examiner, thanks to the helpful recommendations provided by the examiner, the applicant modified the claims with the following changes after consulted with patent agent:

As it seems that claim 1 being too broad, it has been converted to be more specific on the concrete methods. Instead of general steps of allowing mobile device to execute a guest operating system from a host operating system, now the claims contains exact steps for that are used to execute a second operating system in current operating system.

This way, the claims, as amended, therefore appears to overcome indefinite and other issues; withdrawal of the rejection of claims under 35 USC 112 is respectfully requested.

Claim Rejections - 35 USC 103

Claim 1 is rejected under 35 U.S.C. 103(a) as being un-patentable over Griffin et al., U.S. Patent application Publication No. 2002/0194241 A1, in view of Ohno et al. U.S. Patent No. 6,715,016 B1.

Regarding claim 1, Griffin teaches a method of allowing a mobile device having a host OS to execute a guest OS, comprising the steps of [Fig.3]:

preparing an image of the guest OS to be executed within said host OS [paragraph 8, guest OS runs as a virtual machine process on the host OS];

packaging said image of guest OS into a native application of said host OS [paragraph 8, virtual machine];

running said native application from said host OS to execute said guest OS, and exiting said guest OS and retruning back to said host OS [Fig.3, steps 305-307, paragraph 42].

Griffin does not teach reserving memories used by the host OS to preserve current state and data of the host OS.

Ohno teaches a computer system for running multiple operating systems that saves the state of the current OS before switching to a second OS[col. 2, lines 4-11].

It would have been obvious to one of ordinary skill in the art to combine the teachings of Griffin an Ohno by modifying Griffin to save the state of the host OS before switching to the guest OS, as taught by Ohno. Griffin teachs that the guest OS runs as a virtual machine process within the host OS, and that it is describable to increase security and isolation between the host OS and guest OS [paragraph 2-4]. The teachings of Ohno would improve the system of Griffin by providing additional isolation between the memory spaces, resulting in the avoidance of data destruction [col. 2, lines 20-27].

In United States Patent Application 20020194241, Griffin, Jonathan teaches performing secure and insecure computing operations in a compartmented operating system.

[0033] A preferred method for running a process will now be described with reference to FIG. 3. The process performs operations, some of which can be trusted to run securely on the host operating system, and some of which are insecure and cannot be trusted.

[0034] In step 301 a host operating system is provided. Suitably, this is the host operating system 22.

[0035] In step 302 a process 23 runs on the host operating system 22 and attempts to perform a first operation. Preferably, the host operating system 22

provides a compartment 24, and the process 23 is contained by the compartment 24.

[0036] At step 303 operations attempted by the process are monitored. Preferably, the operation attempted by the process is compared with a list of operations falling into a first set and a second set.

[0004] It is known to provide a virtual machine to run on the host operating system. The virtual machine usually includes a guest operating system. The guest operating system commonly is a replica of a second type of operating system such that the computing platform using the host operating system can run processes which are native to the second operating system. The process runs on the guest operating system, and is isolated from the host operating system by the virtual machine. Whilst reasonably secure due to this level of isolation, a virtual machine has a high degree of overhead. In a practical computing environment, it is very inefficient to run a large number of processes on a guest operating system.

[0042] Optionally, in step 307 the guest operating system 25 is closed by the host operating system 22. Suitably, the guest operating system 25 is closed in response to a condition arising as a result of executing the attempted operation on the quest operating system. That is, where the process attempts an untrusted or insecure operation and a dangerous condition arises, then the guest operating system 25 is closed down. Preferably, the host operating system 22 simply closes the relevant compartment 24 containing the guest operating system 25.

According to the discussion with the examiner, Claim 1-19 have been cancelled and substituted with new claims focusing on the additional steps as how to run a guest OS from within a host OS in mobile devices while keeping current the state and data of host OS in memory in an efficient way and minimize the changes to both OS for mobile devices with limited memory and CPU power.

The methods disclosed in Griffin's patent application utilizing a virtual machine and creates a compartment environment for a second OS to be secure from the host OS. As our focus is to allow mobile devices to start a guest OS with minimum data destruction (thus keeping the running state and data of host OS in memory) because in majority of mobile devices, memories and CPU power are limited and traditional OS loading and virtual machine concept and techniques, which requires huge amount of memory and CPU usage, renders impractical.

Furthermore, Griffin's method does not teach exactly steps on how the guest OS can run on top of host OS and then have an efficient way to resume to host OS as users using mobile devices normally expect instant responses. The amended claims aim to provide distinguishable methods as How it is accomplished.

The Ohno, et al. patent has also been carefully reviewed.

"Owing to this configuration, the individual memory spaces for the kernel and the programs are so defined that the interference among OS's such as data destruction may be avoided and that the necessary data may be shared by the programs each executed on the difference OS's."

"In responsive to the generation of interrupt or the request signal from OS's or the software programs running on OS's, this inter-OS control software stores and revises the context information of CPU operations (for example, register values of CPU), switches the memory spaces and restarts the OS operations in another context stored in past".

The applicant contends that the claims as amended are allowable for two independent reasons:

1. The methods disclosed by Ohno requires an inter-OS control software for switching OS and in order to save states, physical memory spaces have to be

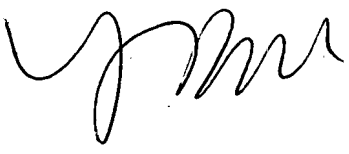
pre-defined and arranged before hand in a particular order. Furthermore, Ohno's methods require two OS each running on its own virtual memory spaces controlled by the inter-OS component which is not practical on many mobile devices due to slow CPU and small memory limitation.

2. Ohno's method mentions about saving and restoring the state of one OS, it does not specifically specify how this can be done. The modified amendments provides detailed steps as how to save the current states of host OS in an efficient way to allow a guest OS being executed on top of the free memories of the host OS.

Conclusion

The applicant has made an earnest effort to present patentably distinguishable claims. In view of the above reasons, it is submitted that new claims 36-54 are now allowable and applicant respectfully request for allowance of the amended claims to such effort. The applicant thanks very much for the work of the examiner, especially to much of the correction and recommendations on applicant's lack of patent practice or skills since the applicant is an individual inventor. The applicant would also greatly appreciate if the respectful examiner could tell if there is any points in the patent applications contain patentable or allowable points in the office action even there are further rejections on those amended claims.

Respectfully submitted,



Yongyong Xu
408-215-8485
yongyongxu@yahoo.com